

# Multidimensional Projection for Uncertain Dataset

Category: Research

**Abstract**— The abstract .....

**Index Terms**—Keyword1, keyword2.

## 1 INTRODUCTION

Introduction to this paper...

## 2 RELATED WORK

( First, we will introduce some conventional MP techniques such as PCA, ISOMap, MDS, LAMP, and so forth and conclude that these approaches can not deal with uncertain datasets. Second, mining uncertain dataset becomes a new ad hoc topic. And several papers will be briefly introduced in this paragraph.)

## 3 OUR APPROACH

This section describes the main concept of our approach. First, each uncertain object is treated as a random variable following a probability distribution. Then, the similarities among these uncertain objects are estimated according to the entropy theory. At last, an MDS algorithm is employed to build a visual representation of uncertain objects which maximally proximate the similarities measured in a high-dimensional space.

### 3.1 Uncertain object and probability distribution

In this paper, each uncertain object which possesses more than one observation is regarded as a random variable following a distribution in a domain  $\mathbb{D}$ . Typically, the exact probability distribution of each random variable is unknown beforehand. However, it can be derived from its observations.

As we know, kernel density estimation(KDE)[7] is a non-parametric method to estimate the probability for a random continuous variable  $X$ . Given a set of observations for a random variable, a kernel function is centered at each observation. And the density at a given position is the sum of influence from all kernels. In practice, a plethora of kernels can be used for estimation such as Gaussian, Tricube, Epanechnikov, and Cauchy kernels. In this paper, Gaussian kernel function is adopted. Finally, a 1D estimator is defined as:

$$p(x) = \frac{1}{n\sqrt{2\pi}h} \sum_{i=1}^m e^{-\frac{(x-X^{(i)})^2}{2h^2}} \quad (1)$$

where  $X^{(i)}$  is the i-th observation of random variable  $X$ ,  $h$  is the bandwidth. The straightforward extension to higher dimensions can be achieved by treating multivariate Gaussian as a product of univariate Gaussians[5]. As a result, a d-dimension estimator is:

$$p(\mathbf{x}) = \frac{1}{m(2\pi)^{d/2} \prod_{k=1}^d h_k} \sum_{i=1}^m \prod_{k=1}^d e^{-\frac{(x_k - X_k^{(i)})^2}{2h_k^2}} \quad (2)$$

here  $X_k^{(i)}$  is the k-th component of an observation  $X^{(i)}$ ,  $h_k$  is the bandwidth at the k-th dimension. For simplicity, a uniform bandwidth is adopted such that Eq. (2) is rewritten as:

$$p(\mathbf{x}) = \frac{1}{m(\sqrt{2\pi}h)^d} \sum_{i=1}^m e^{-\frac{(\mathbf{x}-\mathbf{X}^{(i)})^2}{2h^2}} \quad (3)$$

### 3.2 Dissimilarity estimation

The dissimilarity (distance) measure between pairs of data instances plays an essential role in most multidimensional projection techniques. And the concept of *Euclidian* distance has prevailed in a large number of multidimensional projection approaches. It is simple and easy to use. However, we can not find a direct definition of *Euclidian* distance for uncertain objects.

A naive way to quantify the dissimilarities among uncertain objects is to simplify the uncertain data to certain data and use the *Euclidian* distances to approximate the dissimilarities. Unfortunately, this simple strategy may not use the full potential of data and would give rise to information loss. Figure 1 shows a simple 2D example. The dataset used in this example is composed of four uncertain objects with the same mean value but different distribution. Evidently, it is impossible to distinguish each uncertain object through a simplifying process that we replace each uncertain object with the mean value.

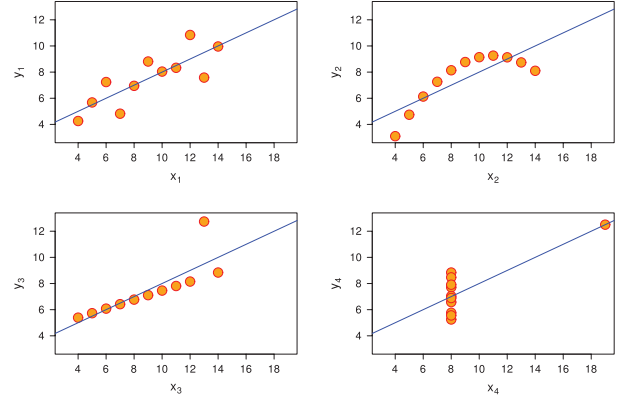


Fig. 1. A simple 2D example: four uncertain objects with the same mean value but different distribution type.

In the field of probability theory and statistics, the *Jensen-Shannon* divergence is a very popular metric of measuring similarity of two probability distributions. It is derived from the *Kullback-Leibler* divergence which is always defined in a discrete domain. In our approach, we simply extend it to a continuous domain as:

$$J(P||Q) = \frac{1}{2} \left( \int_{\mathcal{D}} p(\mathbf{x}) \log \frac{2p(\mathbf{x})}{p(\mathbf{x})+q(\mathbf{x})} d\mathbf{x} + \int_{\mathcal{D}} q(\mathbf{x}) \log \frac{2q(\mathbf{x})}{p(\mathbf{x})+q(\mathbf{x})} d\mathbf{x} \right) \quad (4)$$

where,  $p(x)$  and  $q(x)$  are the probability distribution function of uncertain object  $P$  and  $Q$  respectively. Typically, we set the base of log as 2 such that  $J(P||Q)$  is bounded by 1. Only  $P$  and  $Q$  have the same distribution does  $J(P||Q)$  become 0. By convention,  $0 \log 0$  is defined as zero. Similarly,  $0 \log \frac{0}{0}$  is treated as zero as well.

Consequently, we derive the dissimilarity between a pair of uncertain objects  $P$  and  $Q$  as the summation of the *Euclidian* distance between their mean values and the *Jensen-Shannon* divergence[3] be-

tween their probability distributions.

$$d(P, Q) = \lambda E(\bar{P}, \bar{Q}) + \theta J(P||Q) \quad (5)$$

Here,  $\bar{P}$  and  $\bar{Q}$  are the mean value of uncertain object  $P$  and  $Q$ ,  $J(P||Q)$  is the *Jensen-Shannon* divergence,  $\lambda$  and  $\theta$  are two user adjustable parameters, and  $\lambda, \theta \in [0, 1]$ . As  $\theta$  goes to zero, our method will degenerate to a conventional multi-dimensional projection technique for certain dataset.

In summary, our derived dissimilarity embraces the following properties:

- The dissimilarity between two objects  $P$  and  $Q$  is symmetric, which permits that it can be seamlessly incorporated with many state-of-the-art multidimensional projection techniques.
- If two objects have similar mean values, they still can be distinguished from each other by the dissimilarity of their probability distributions.

### 3.3 Dissimilarity preserving embedding

## 4 IMPLEMENTATION

### 4.1 Accelerating probability density estimation

As we can see that the probability distribution for each individual uncertain object can be directly calculated by Eq. (3). However, the complexity highly depends on the number of observations and the dimensionality of the uncertain object. Of course, an offline pre-computation process can be utilized. But this would require much more disk resources to store the pre-computation results because of the infinite integration domain  $\mathbb{D}$ .

In fact, given  $n$  uncertain objects with  $m$  observations for each uncertain object, the direct evaluation complexity is  $O(n \times m)$ . So, bigger  $n$  and/or  $m$  would lead to sever requirements for computation time and storage. To make our approach practical, we adopt an advanced technique called *Improved Fast Gauss Transformation (IFGT)*[8] to accelerate this process.

The core idea of *IFGT* is based on the fact that the Gaussian, especially in higher dimensions, decays rapidly so that the contribution outside of a certain region can be ignored. And it uses the following truncated multivariate Taylor expansion for factorization.

$$e^{-\frac{(\mathbf{x}-\mathbf{x}^{(i)})^2}{2h^2}} = e^{-\frac{(\mathbf{x}^{(i)}-\mathbf{x}^{(*)})^2}{2h^2}} e^{-\frac{(\mathbf{x}-\mathbf{x}^{(*)})^2}{2h^2}} Y_\alpha \quad (6)$$

$$Y_\alpha = \sum_{|\alpha| \leq t-1} \frac{2^{|\alpha|}}{\alpha!} \left( \frac{\mathbf{x}^{(i)} - \mathbf{x}^{(*)}}{\sqrt{2}h} \right)^\alpha \left( \frac{\mathbf{x} - \mathbf{x}^{(*)}}{\sqrt{2}h} \right)^\alpha \quad (7)$$

Where  $\mathbf{x}^{(*)}$  is the expansion center,  $t$  is the truncation degree, and  $\alpha$  is multi-index notation.

Finally, our acceleration process involves the following steps:

**Step 1** Partition the  $m$  observation for an uncertain object into  $K$  clusters such that the radius is less than  $h\rho_1$  by using techniques such as farthest-point clustering[4]. Here,  $\rho_1$  is a controlling parameter. The cluster centers will be used as the expansion centers later.

**Step 2** Choose a big enough truncation degree  $t$  such that the total error is bounded by a given precision  $\varepsilon$ .

**Step 3** For each cluster center  $C_k$ , calculate the constant coefficients of Taylor expansion.

$$C_{\alpha,k} = \frac{2^{|\alpha|}}{m(\sqrt{2\pi}h)^d \alpha!} \sum_{i=1}^m e^{-\frac{(\mathbf{x}^{(i)}-C_k)^2}{2h^2}} \left( \frac{\mathbf{x}^{(i)} - C_k}{\sqrt{2}h} \right)^\alpha \quad (8)$$

**Step 4** For each sample point  $\mathbf{x} \in \mathbb{D}$ , find its neighbor cluster centers with radius less than  $h\rho_2$  and compute the probability  $p(\mathbf{x})$  by:

$$p(\mathbf{x}) = \sum_{(\mathbf{x}-C_k)^2 \leq h\rho_2} \sum_{\alpha < t} C_{\alpha,k} e^{-\frac{(\mathbf{x}-C_k)^2}{2h^2}} \left( \frac{\mathbf{x} - C_k}{\sqrt{2}h} \right)^\alpha \quad (9)$$

### 4.2 Computing Jensen-Shannon divergence

In continuous case, it is very difficult to evaluate Eq. (4) to compute the *Jensen-Shannon* divergence between  $P$  and  $Q$ . However, with the law of large number, we have:

$$J(P||Q) = \frac{1}{2} \left( \sum_{i=1}^{s_1} \log \frac{2p(\mathbf{P}^{(i)})}{p(\mathbf{P}^{(i)}) + q(\mathbf{P}^{(i)})} + \sum_{i=1}^{s_2} \log \frac{2q(\mathbf{Q}^{(i)})}{p(\mathbf{Q}^{(i)}) + q(\mathbf{Q}^{(i)})} \right) \quad (10)$$

where  $\{\mathbf{P}^{(1)}, \mathbf{P}^{(2)}, \dots, \mathbf{P}^{(s_1)}\} \sim p(\mathbf{x})$ ,  $\{\mathbf{Q}^{(1)}, \mathbf{Q}^{(2)}, \dots, \mathbf{Q}^{(s_2)}\} \sim q(\mathbf{x})$ ,  $s_1$  and  $s_2$  are the sample size. To make sure that the samples  $\mathbf{P}_i$  and  $\mathbf{Q}_i$  mimic samples drawn from the target distributions  $p(\mathbf{x})$  and  $q(\mathbf{x})$ , we use the *Metropolis-Hastings* algorithm[1] which is the most popular *Markov chain Monte Carlo* (MCMC) method to generate samples.

In practice, we can further accelerate the computation by a simple trick. For convenience, we refer to the subspace satisfying  $p(\mathbf{x}) > 0$  and  $q(\mathbf{x}) > 0$  as valid integration region  $\mathbb{D}$ . Obviously, if  $\mathbb{D}$  is empty, we can simply set the *Jensen-Shannon* divergence between these objects as 1 without any sampling. This assumption can be easily proved by Eq. (10). And to check whether  $\mathbb{D}$  is empty can be regarded as a process to testify whether two high-dimensional bounding boxes are overlapped. Here, a high-dimensional bounding box of an uncertain object records the lowest and hight values of all observations associated with this object.

## 5 RESULTS AND DISCUSSIONS

We implemented our approach with C++ and conducted a large amount of experiments on both synthetic datasets and real datasets on a PC equipped with an Intel Core 2 Duo 3.0 GHz CPU, 4GB host memory and an Nvidia GTX580 video card with 1GB video memory.

### 5.1 Generating synthetic dataset

To verify our approach, we followed the way that Jiang et al.[6] used to generate our synthetic datasets. Roughly speaking, the generation process is controlled by 4 parameters: the dimensionality  $d$  of the data space; the number of uncertain objects  $n$ ; the number of observations  $m$  for each uncertain object; and the number of clusters  $k$ . Uncertain objects belong to a same cluster  $i$  have the same mean value  $\mu_i = 0.05i \frac{n}{k}$  and variance  $v_i = 0.5i$ , but with different distributions. In this paper, three different types of distribution including the uniform distribution, the Gaussian distribution, and the inverse Gaussian distribution are adopted. Finally, the  $j$ -th uncertain object is created by setting its cluster as  $C_j \equiv j \bmod k$  and generating  $m$  samples following a specific distribution randomly chosen from the three above.

### 5.2 Results with synthetic dataset

Figure 2 shows the results of a synthetic 5D dataset consisting of 100 uncertain objects. And each uncertain object has 1000 observations. Figure 2(a) denotes the result of applying conventional MDS[2] on the simplified certain dataset through replacing each uncertain object with its mean value. Figure 2(b) is the result of our approach. The color of each point in this example encodes the distribution type. Specifically, red is for uniform distribution, green for Gaussian distribution, and blue for inverse Gaussian distribution.

Compared with the conventional MDS, our method can also preserve the global data structure, in this example, that is the linearity which is encoded in the process of generating uncertain dataset. In addition, since the statistical differences are taken into account in our method, the resulting projection of our method is much more accurate than that of the conventional MDS. As illustrated in the zoomed region(indicated by the brown rectangles), our method gains much more insight of the dataset. The objects in the same cluster nearly having an identical mean value are mixed together in the 2D space by applying the conventional MDS algorithm. But they can be easily distinguished from each other with our approach. 2(b) depicted, objects with similar distributions will attract each other, while objects with different distributions will repel each other.

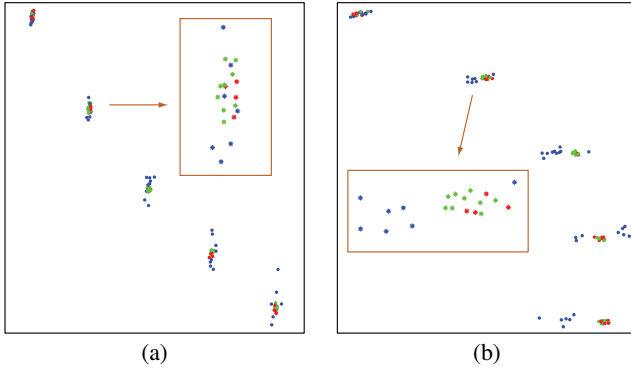


Fig. 2. Results of synthetic dataset: (a) result of applying conventional MDS; (b) our result with  $\lambda = 0.85, \theta = 0.5$

### 5.3 Results with real dataset

## 6 CONCLUSIONS

### REFERENCES

- [1] C. Andrieu, N. De Freitas, A. Doucet, and M. Jordan. An introduction to mcmc for machine learning. *Machine learning*, 50(1):5–43, 2003.
- [2] I. Borg and P. Groenen. *Modern multidimensional scaling: Theory and applications*. Springer Verlag, 2005.
- [3] S.-H. Cha. Comprehensive survey on distance/similarity measures between probability density functions. *International Journal of Mathematical Models and Methods in Applied Sciences*, 1(4):300–307, 2007.
- [4] T. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.
- [5] L. Greengard and J. Strain. The fast gauss transform. *SIAM J. Sci. Stat. Comput.*, 12(1):79–94, 1991.
- [6] B. Jiang, J. Pei, Y. Tao, and X. Lin. Clustering uncertain data based on probability distribution similarity. *Knowledge and Data Engineering, IEEE Transactions on*, (99):1–1, 2011.
- [7] B. Silverman. *Density estimation for statistics and data analysis*, volume 26. Chapman and Hall/CRC, 1986.
- [8] C. Yang, R. Duraiswami, N. A. Gumerov, and L. Davis. Improved fast gauss transform and efficient kernel density estimation. In *ICCV*, pages 464–471, 2003.